May 2nd, 11:00 AM - 1:00 PM

Using Reservoir Computing to Build a Robust Interface with DNA Circuits in Determining Genetic Similarities Between Pathogens

Christopher Neighbor
Portland State University

Christof Teuscher
Portland State University, teuscher@pdx.edu

Using Reservoir Computing to Build a Robust Interface with DNA Circuits in Determining Genetic Similarities Between Pathogens

Christopher Neighbor, PI: Dr. Christof Teuscher
Teuscher Lab, Portland State University

teuscher.:Lab

Portland State UNIVERSITY

Abstract

As computational power increases, the field of neural networks has advanced exponentially. In particular recurrent neural networks (RNNs) are being utilized to simulate dynamic systems and to learn to predict time series data. Reservoir computing is an architecture which has the potential to increase training speed while reducing computational costs. Reservoir computing consists of a RNN with a fixed connections "reservoir" while only the output layer is trained. The purpose of this research is to explore the effective use of reservoir computing networks with the eventual application towards use in a DNA based molecular computing reservoir for use in pathogen detection.

Objectives

An end goal vision is to create a robust medical pathogen test, as easy to use as a pregnancy test, which is able to detect new unknown mutations of known pathogens.

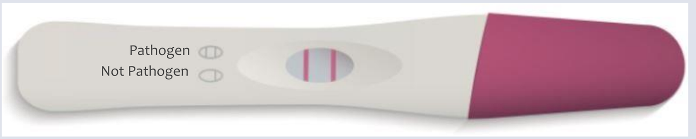

Image: Shutterstock

Other goals of the this research include:

- Develop an effective and predictable interface between molecular circuits and biological systems
- Simulate molecular circuit architectures in order to design a more robust architecture which is more resilient to interference and increased scale and complexity
- Use genomic databases to train a reservoir network to identify mutations of a particular pathogen

Here at PSU, I am focused on experimenting with these networks as purely computer simulation, or *in silico*, in order to determine potential optimizations in data preprocessing, format of the input streams, network architecture, and output layer training. The wet chemistry applications of this research will be carried out at University of New Mexico.

Methods

As an initial step, a neural network is being trained to measure the Hamming distance between two binary input streams; the Hamming distance is simply the number of bits between the two inputs which are different (e.g. 1001 and 1110 have a Hamming distance of 3). It is commonly used in error correction code. In this research it is used as an initial measurement of the difference between two genetic data input streams, thus, a genetic Hamming distance (e.g. ATCCG and ATTGC has a genetic Hamming distance of 3). A simplistic model could classify genetic strands which have a smaller genetic Hamming distance as being more likely to be the same pathogen.

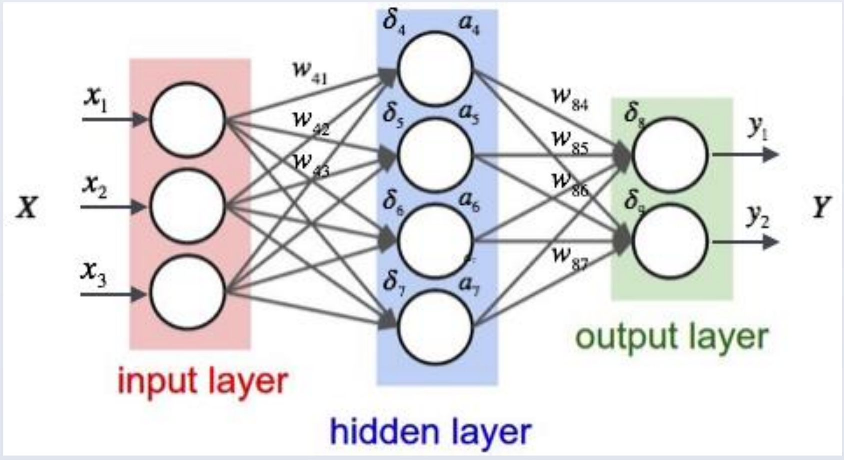

Figure 1. Structure of a typical feed forward neural net with forward propagation [1]

The primary architecture to be explored in this research is reservoir computing, a form of recurrent neural net in which only the output layer is trained.

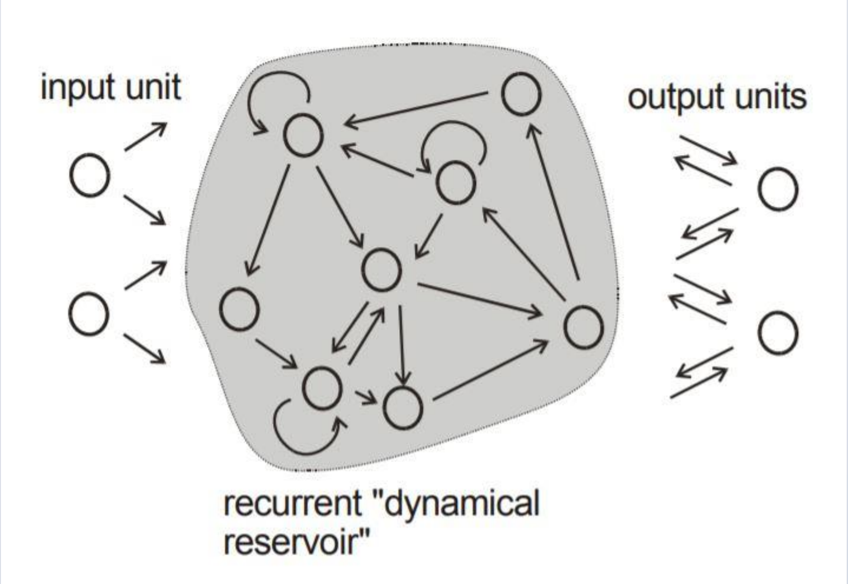

Figure 2. Schematic of a recurrent reservoir architecture[2]

For my initial project, I used a trained RNN built using long short-term memory (LSTM) cells which allow the network to have a memory of the previous inputs and outputs of the system.

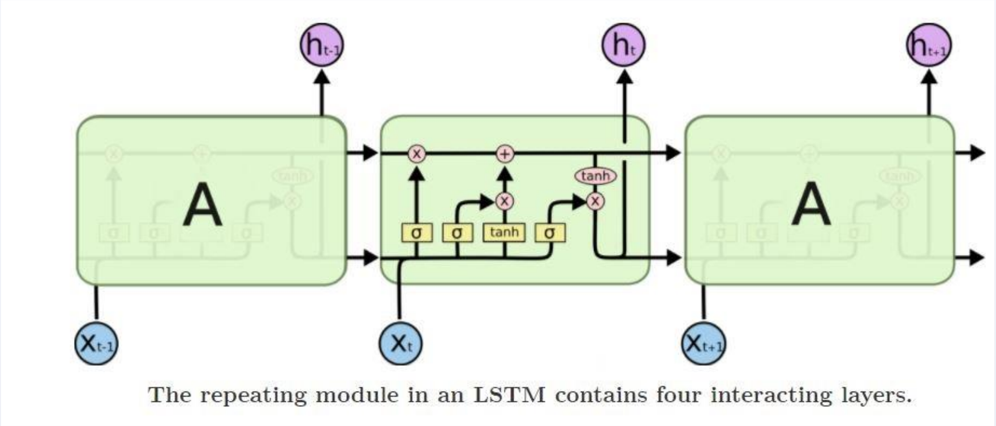

Figure 3. Diagram of the LSTM node used in the RNN allowing for gated memory[3]

The use of an LSTM allows for control of memory from previous time points in order to effectively predict values at future time points.

For training the network, a dataset consisting of two binary streams was created with the actual Hamming distance information for its supervised learning.

Results

The RNN network was able to be trained using stochastic gradient descent and backpropagation through time in order to update the weights of the network and fit itself to the labeled training dataset.

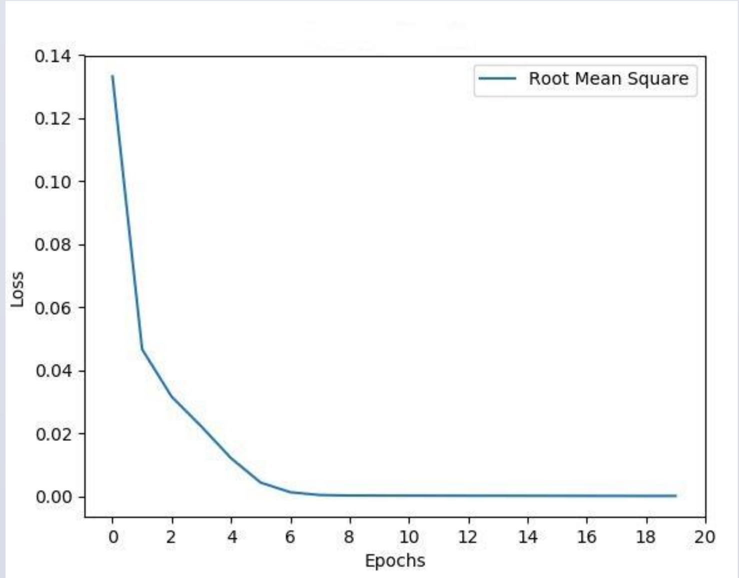

Figure 3. Error of the model as the RNN increased its training iterations

$$RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(expected\ ouput_n - predicted\ ouput_n)^2}$$

Equation 1. Root mean square is the loss function being minimized by the neural network

Using this trained network, I was able to generate a time-series predictive model which was able to reduce its loss function and model the continuing Hamming distance sequence for the test dataset.

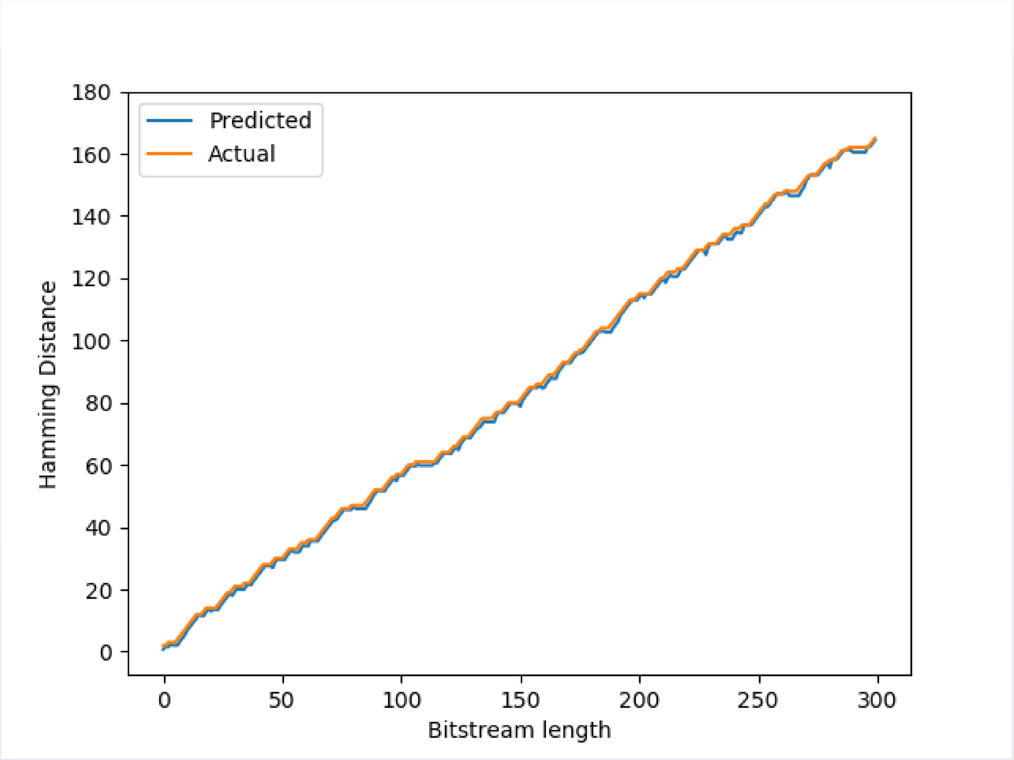

Figure 4. Using the trained RNN to predict the Hamming distance between two binary input streams.

A small trained RNN can closely predict the Hamming distance between two binary input streams. Improvements can still be made and increased parameter tuning could increase this models performance in this task.

Conclusions and Future Work

Initial results with a trained RNN have been promising in its ability to learn an effective prediction of Hamming distance between two given binary input streams with only a short-term memory.

The advantage of construction of these networks *in silico* is that it allows for easy and rapid changing of both its architecture and its training parameters. The next step in my project is to create an untrained RNN, the reservoir, and train just the output layer to learn the Hamming distance and effective classification of pathogens based on their genetic sequence.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant no. 1518833

References

[1]Valkov, V. (2017). *Creating a Neural Network from Scratch—TensorFlow for Hackers (Part IV)*. [online] Medium. Available at: https://medium.com/@curiousily/tensorflow-for-hackers-part-iv-neural-network-from-scratch-1a4f504dfa8 [Accessed 16 Apr. 2018

[2]Jaeger, H. (2016). *A quick introduction to reservoir computing*. [online] Minds.jacobs-university.de. Available at: http://minds.jacobs-university.de/sites/default/files/uploads/ teaching/ MLSpring16/ReservoirComputing.pdf [Accessed 15 Apr. 2018].

[3]Olah, C. (2015). *Understanding LSTM Networks -- colah's blog*. [online] Available at: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ [Accessed 15 Apr. 2018].